1. There are n cash desks in a big store, k customers select simultaneously and independently a cash desk at random, so the selection is done with equal probabilities. Please find the expectation of the number of busy cash desks.

2. Given an $M \times N$ table with numbers from 1 to K written in each cell, $K \leq M \cdot N$. Each number is written in at least one cell. A pair of numbers (x, y), $1 \leq x < y \leq K$, is called a neighbor, if there is at least one pair of cells marked with these numbers such that they share an edge (i.e., they are horizontal or vertical neighbors) in the table. What lower bound for the number of such neighbors can you provide? Is it exact?

3. A cluster consists of N servers. The server i, $1 \leq i \leq N$, can serve up to $C_i \in \mathbb{N}$ virtual machines (VM). We are given M virtual machines. A VM m, $1 \leq m \leq M$, can be allocated on one of two servers $\{i_m, j_m\}$ (the list of pairs is given in the input). The number of allocated VMs should be maximized. Write a program that allocates VMs and meets the mentioned requirements. For testing, you also need to write a program that generates a random examples of input data for the given N, M and $\sum_{1 \leq i \leq N} C_i = M$. The input has to be written in two CSV-files: `servers.csv` has two columns `id,capacity`; `VM.csv` has 3 columns `id,server1,server2`. The output has to be written in the file `allocated.csv` with two columns `VM,server`.

4*. Given a tree T with non-negative weights of edges and non-positive weights of nodes. The weight of a subgraph $H \subseteq T$ is the sum of weights of its nodes and edges. Consider a problem of finding a subgraph of maximum weight. Develop and implement an efficient algorithm for solving this problem. Also, prepare a code that will generate a random tree with random weights (small integers) of nodes and edges. Use it for testing your algorithm. The input has to be written in two CSV-files: `nodes.csv` has two columns `id,weight`; `edges.csv` has 4 columns `id,node1,node2,weight`. The output has to be written in the file `subgraph.csv` with one column `edge`.